

## CLAIMS

What is claimed is:

1. A computer system comprising:
  - a processor;
  - a memory device;
  - a CAM coupled to the processor to store requests for write transactions received from the processor as pending write transactions;
  - a first latch coupled to the CAM to store an address of a currently selected write transaction received from the CAM;
  - transaction logic coupled to CAM and to the memory device to carry out the currently selected write transaction received from the CAM;
  - a second latch coupled to the first latch to store a copy of the address of the currently selected write transaction received from the first latch at a subsequent time when another pending write transaction is selected, thereby causing the currently selected write transaction to be replaced and to become the previously selected write transaction;
  - a multiplexer coupled to the outputs of both the first and second latches and to the CAM to select between an addresses stored in the first and second latches to become the comparand input to the CAM for a comparison; and
  - control logic coupled to CAM to receive the results of the comparison and further coupled to the transaction logic to selectively accompany a write transaction selected from the CAM and sent to the transaction logic with a page closing hint signal to the transaction logic.

2. The computer system of claim 1, further comprising:  
a read transaction buffer to store pending read transactions; and  
arbiter logic coupled to the output of the read transaction buffer and the CAM to make a determination whether a read transaction or a write transaction is to be the next transaction to be carried out, and further coupled to the transaction logic to supply the transaction logic with the result of the determination.
3. The computer system of claim 1, further comprising mask logic coupled to the multiplexer to screen out bits correlating to a column of memory cells within a page such that the comparison does not include the specification of a column, and to selectively and additionally screen out bits correlating to a page such that the comparison selectively does not include the specification of a page.
4. The computer system of claim 3, wherein the address of a pending write transaction stored within the write transaction buffer is stored in the form of a single binary value specifying the starting address of a portion of the memory device to which the write transaction is directed, and further comprising conversion logic to convert the single binary value to a plurality of values that separately specify at least a bank and a page, wherein the associative comparison receives the plurality of values as an input.
5. The computer system of claim 1, further comprising a nonvolatile portion of memory within the memory device indicating the configuration of ranks, banks, pages and columns into which memory cells within the memory device are organized.
6. The computer system of claim 5, wherein the nonvolatile portion of memory within the memory device is accessible by the processor, and wherein the mask logic is programmable by the processor to permit the processor to program the mask logic with

mask values based on the configuration of ranks, banks, pages and columns indicated by the nonvolatile portion of the memory within the memory device.

7. The computer system of claim 1, wherein the transaction logic transmits a write command to the memory device along with an auto precharge command if a write transaction selected from the CAM and sent to the transaction logic is accompanied with a page closing hint signal.

8. A memory controller comprising:

- a transaction buffer to store pending write transactions;
- a first latch coupled to the output of the transaction buffer to store an address of a currently selected write transaction received from the transaction buffer;
- transaction logic coupled to the output of the transaction buffer to carry out the currently selected write transaction received from the transaction buffer;
- a second latch coupled to the first latch to store a copy of the address of the currently selected write transaction received from the first latch at a time when another pending write transaction selected subsequently, thereby causing the currently selected write transaction to be replaced and to become the previously selected write transaction;
- a multiplexer coupled to the outputs of both the first and second latches to select between an address stored in the first latch and an address stored in the second latch to become a comparand;
- comparison logic coupled to the output of the multiplexer and to the transaction buffer to carry out an associative comparison between the comparand received from the multiplexer and addresses of the pending write transactions stored in the transaction buffer; and
- control logic coupled to the output of the comparison logic to receive the results of the associative comparison and further coupled to the transaction logic to selectively

accompany a write transaction selected from the transaction buffer and sent to the transaction logic with a page closing hint signal to the transaction logic.

9. The memory controller of claim 8, further comprising:  
a separate read transaction buffer to store pending read transactions; and  
arbiter logic coupled to the outputs of both the separate read transaction buffer and the transaction buffer to make a determination whether a read transaction or a write transaction is to be the next transaction to be carried out, and further coupled to the transaction logic to supply the transaction logic with the result of the determination.

10. The memory controller of claim 8, wherein the transaction buffer also stores read transactions, and further comprising arbiter logic coupled to the output of the transaction buffer to make a determination whether a read transaction or a write transaction is to be the next transaction to be carried out, and further coupled to the transaction logic to supply the transaction logic with the result of the determination.

11. The memory controller of claim 8, further comprising mask logic coupled to the multiplexer to screen out bits correlating to a column of memory cells within a page such that the associative comparison does not include the specification of a column, and to selectively and additionally screen out bits correlating to a page such that the associative comparison selectively does not include the specification of a page.

12. The memory controller of claim 11, wherein the address of a pending write transaction stored within the write transaction buffer is stored in the form of a single binary value specifying the starting address of a portion of memory coupled to the memory controller and to which the write transaction is directed, and further comprising conversion logic to convert the single binary value to a plurality of values that

separately specify at least a bank and a page, wherein the associative comparison receives the plurality of values as an input.

13. The computer system of claim 8, wherein the transaction logic transmits a write command to the memory device along with an auto precharge command if a write transaction selected from the write transaction buffer and sent to the transaction logic is accompanied with a page closing hint signal.

14. A method comprising:

making available a first storage location for storing the address to which a new selected\_entry write transaction will be directed while continuing to store the address to which a recent selected\_entry write transaction was directed, thereby causing the recent selected\_entry write transaction to become the last\_write\_selected write transaction;

associatively comparing at least the bank and page of the last\_write\_selected write transaction to at least the bank and page of pending write transactions stored in a transaction buffer;

selecting a pending write transaction from within the transaction buffer to become the new selected\_entry write transaction, removing the selected\_entry write transaction from the transaction buffer, and provide the selected\_entry write transaction with a page closing hint to transaction logic to carry out the selected\_entry write transaction if the selected\_entry write transaction was the only write transaction in the transaction buffer with at least the same bank and page as the last\_write\_selected write transaction; and

selecting a pending write transaction from within the transaction buffer to become the new selected\_entry write transaction, removing the selected\_entry write transaction from the transaction buffer, and provide the selected\_entry write transaction without a page closing hint to transaction logic to carry out the selected\_entry write

transaction if the selected\_entry write transaction was one of multiple pending write transactions in the transaction buffer with at least the same bank and page as the last\_write\_selected write transaction.

15. The method of claim 14, further comprising:

selecting a pending write transaction from within the transaction buffer to become the new selected\_entry write transaction, removing the selected\_entry write transaction from the transaction buffer, associatively comparing at least the bank and page of the selected\_entry write transaction to at least the bank and page of the remaining pending write transactions stored in the transaction buffer, providing the selected\_entry write transaction with a page closing hint to transaction logic to carry out the selected\_entry write transaction if there were no pending write transactions found in the transaction buffer directed to at least the same bank and page as the last\_write\_selected transaction, the selected\_entry write transaction was directed to a different bank than the last\_write\_selected write transaction, and no other pending write transaction in the transaction buffer is directed to the same bank and page as the selected\_entry write transaction; and

selecting a pending write transaction from within the transaction buffer to become the new selected\_entry write transaction, removing the selected\_entry write transaction from the transaction buffer, associatively comparing at least the bank and page of the selected\_entry write transaction to at least the bank and page of the remaining pending write transactions stored in the transaction buffer, providing the selected\_entry write transaction without a page closing hint to transaction logic to carry out the selected\_entry write transaction if there were no pending write transactions found in the transaction buffer directed to at least the same bank and page as the last\_write\_selected transaction, the selected\_entry write transaction was directed to a different bank than the last\_write\_selected write transaction, and there is at least one

other pending write transaction in the transaction buffer directed to the same bank and page as the selected\_entry write transaction.

16. The method of claim 14, further comprising:

selecting a pending write transaction from within the transaction buffer to become the new selected\_entry write transaction, removing the selected\_entry write transaction from the transaction buffer, associatively comparing at least the bank and page of the selected\_entry write transaction to at least the bank and page of the remaining pending write transactions stored in the transaction buffer, providing the selected\_entry write transaction with a page closing hint to transaction logic to carry out the selected\_entry write transaction if there were no pending write transactions found in the transaction buffer directed to at least the same bank and page as the last\_write\_selected transaction, there were no pending write transactions found in the transaction buffer directed to a different bank than the last\_write\_selected transaction, the selected\_entry write transaction was directed to the same bank as the last\_write\_selected write transaction, but to a different page than the last\_write\_selected write transaction, and no other pending write transaction in the transaction buffer is directed to the same bank and page as the selected\_entry write transaction; and

selecting a pending write transaction from within the transaction buffer to become the new selected\_entry write transaction, removing the selected\_entry write transaction from the transaction buffer, associatively comparing at least the bank and page of the selected\_entry write transaction to at least the bank and page of the remaining pending write transactions stored in the transaction buffer, providing the selected\_entry write transaction without a page closing hint to transaction logic to carry out the selected\_entry write transaction if there were no pending write transactions found in the transaction buffer directed to at least the same bank and page as the last\_write\_selected transaction, there were no pending write transactions found in the

transaction buffer directed to a different bank than the last\_write\_selected transaction, the selected\_entry write transaction was directed to the same bank as the last\_write\_selected write transaction, but to a different page than the last\_write\_selected write transaction, and there is at least one other pending write transaction in the transaction buffer is directed to the same bank and page as the selected\_entry write transaction.